|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ «Информатика и системы управления»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Лабораторная работа № \_\_3\_\_**

|  |  |
| --- | --- |
| **Тема:** Программно- алгоритмическая реализация моделей на основе ОДУ второго порядка с краевыми условиями II и III рода  **Студент \_\_\_\_**Оберган Т.М.**\_\_\_\_\_\_**  **Группа \_\_\_\_**ИУ7-65Б\_\_\_\_\_\_\_\_\_\_\_  **Оценка (баллы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Преподаватель \_\_\_**Градов В.М.\_\_\_ |  |

Москва.

2020 г.

**Оглавление**

[Исходные данные 3](#_Toc39097287)

[Физическое содержание задачи 5](#_Toc39097288)

[Листинг 6](#_Toc39097289)

[Результаты работы 7](#_Toc39097290)

[Вопросы при защите лабораторной работы 12](#_Toc39097291)

**Цель работы**: получение навыков разработки алгоритмов решения краевой задачи при реализации моделей, построенных на ОДУ второго порядка.

# Исходные данные

1. Задана математическая модель.

Уравнение для функции :

(1)

Краевые условия:

(2)

2. Функции заданы константами:

(3,4)

Константы следует найти из условий , а константы из условий . Получим:

(5-8)

Величины задает пользователь.

3. Разностная схема с разностным краевым условием при .

(9)

Система (9) совместно с краевыми условиями решается методом прогонки.

Для величин можно получить различные приближенные выражения, численно вычисляя интеграл методом трапеций или методом средних.

Для вычислений будет использоваться метод средних:

(10)

Разностный аналог краевого условия при :

(11)

Простая аппроксимация:

(12,13)

При выполнении лабораторной необходимо учесть, что

(14,15)

4. Значения параметров для отладки (все размерности согласованы)

0.4 Вт/см К,

0.1 Вт/см К,

0.05 Вт/см2 К,

0.01 Вт/см2 К,

 10 см,

300К,

0.5 см,

50 Вт/см2.

# Физическое содержание задачи

Сформулированная математическая модель описывает температурное поле  вдоль цилиндрического стержня радиуса  и длиной , причем  и температуру можно принять постоянной по радиусу цилиндра. Ось  направлена вдоль оси цилиндра и начало координат совпадает с левым торцом стержня. Слева при  цилиндр нагружается тепловым потоком . Стержень обдувается воздухом, температура которого равна . В результате происходит съем тепла с цилиндрической поверхности и поверхности правого торца при .Функции являются, соответственно, коэффициентами теплопроводности материала стержня и теплоотдачи при обдуве,.

# Листинг

# Лабораторная по моделированию №3

# Реализация модели на основу ОДУ второго

# порядка с краевыми условиями II и III рода.

**import** matplotlib.pyplot **as** plt

**import** numpy **as** np

# Исходные данные модели

**def** get\_constants():

b = (kN \* N) / (kN - k0)

d = (aN \* N) / (aN - a0)

a = -k0 \* b

c = -a0 \* d

**return** a, b, c, d

**def** k(x):

**return** a / (x - b)

**def** alpha(x):

**return** c / (x - d)

**def** p(x):

**return** (2 \* alpha(x)) / R

**def** f(x):

**return** (2 \* T0 \* alpha(x)) / R

# Разностная схема

**def** A(n):

**return** approc\_plus\_half(k, n) / h

**def** C(n):

**return** approc\_minus\_half(k, n) / h

**def** B(n):

**return** A(n) + C(n) + p(n) \* h

**def** D(n):

**return** f(n) \* h

# Простая аппроксимация

**def** approc\_plus\_half(func, n):

**return** (func(n) + func(n + h)) / 2

**def** approc\_minus\_half(func, n):

**return** (func(n - h) + func(n)) / 2

# Краевые условия

# При х = 0

**def** left\_boundary\_condition():

k0 = approc\_plus\_half(k, 0) + (h\*h\*approc\_plus\_half(p, 0) / 8) + ((h\*h\*p(0)) / 4)

M0 = -approc\_plus\_half(k, 0) + (h\*h\*approc\_plus\_half(p, 0) / 8)

P0 = h\*F0 + (h\*h / 4) \* (approc\_plus\_half(f, 0) + f(0))

**return** k0, M0, P0

# При x = N

**def** right\_boundary\_condition():

kN = (approc\_minus\_half(k, N) / h) - (h\*approc\_minus\_half(p, N)/ 8)

MN = -aN - (approc\_minus\_half(k, N) / h) - (h\*p(N) / 4) - (h\*approc\_minus\_half(p, N) / 8)

PN = -(h/4) \* (f(N) + approc\_minus\_half(f, N)) - T0\*aN

**return** kN, MN, PN

**if** \_\_name\_\_ == "\_\_main\_\_":

# Исходные данные

k0 = 0.4

kN = 0.1

a0 = 0.05

aN = 0.01

N = 10

T0 = 300

R = 0.5

F0 = 50

h = 1***e***-3

a, b, c, d = get\_constants()

k0, M0, P0 = left\_boundary\_condition()

kN, MN, PN = right\_boundary\_condition()

# Прямой ход

eps = [0, -M0 / k0]

eta = [0, P0 / k0]

x = h

n = 1

**while** x + h < N:

eps.append(C(x) / (B(x) - A(x) \* eps[n]))

eta.append((A(x) \* eta[n] + D(x)) / (B(x) - A(x) \* eps[n]))

n += 1

x += h

# Обратный ход

t = [0] \* (n + 1)

t[n] = (PN - MN \* eta[n]) / (kN + MN \* eps[n])

**for** i **in** **range**(n - 1, -1, -1):

t[i] = eps[i + 1] \* t[i + 1] + eta[i + 1]

# График

x = [i **for** i **in** np.arange(0, N, h)]

plt.plot(x, t[:-1], 'r-')

plt.xlabel("x, cm")

plt.ylabel("temperature, K")

plt.grid()

plt.show()

# Результаты работы

1. Представить разностный аналог краевого условия при  и его краткий вывод интегро-интерполяционным методом.

Пусть

Тогда (1) можно записать как

(16)

Проинтегрируем (15) на отрезке :

(17)

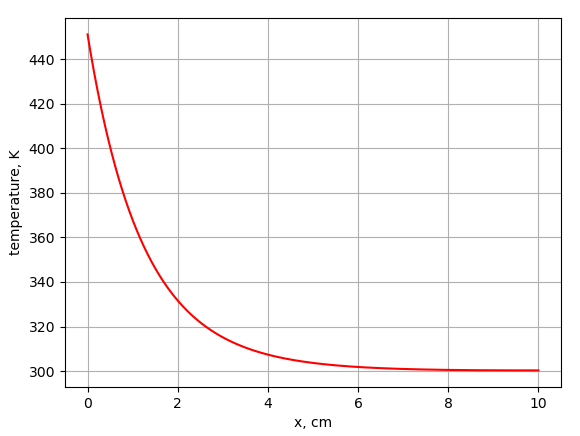
Второй и третий интегралы вычислим методом трапеций:

(18)

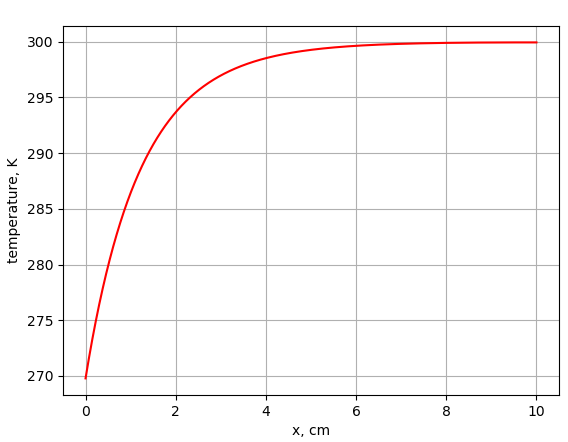
Применив (14, 15) к (18) получим:

(19)

2. График зависимости температуры  от координаты  при заданных выше параметрах.

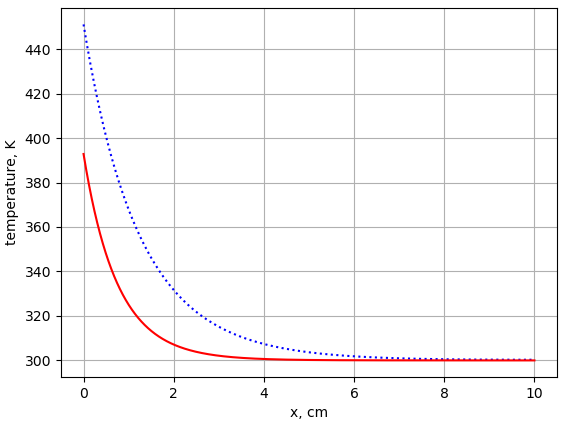


3. График зависимости  при -10 Вт/см2.



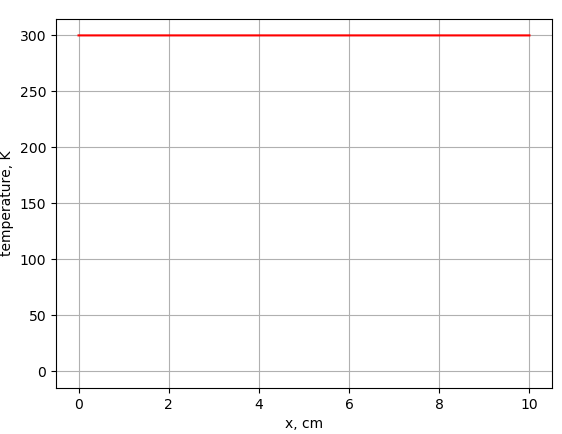
*Справка*. При отрицательном тепловом потоке слева идет съем тепла, поэтому производная  должна быть положительной.

4. График зависимости  при увеличенных значениях (например, в 3 раза). Сравнить с п.2.

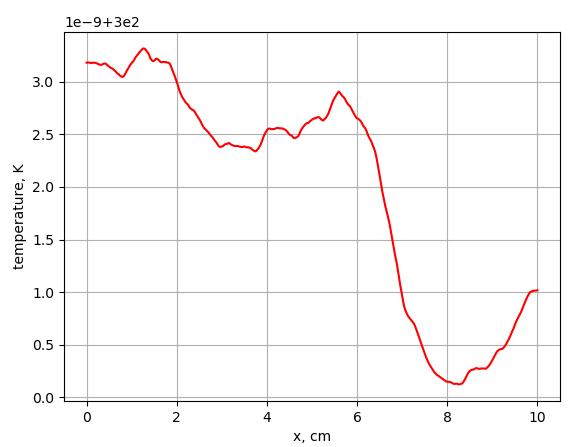


*Справка*. При увеличении теплосъема и неизменном потоке  уровень температур  должен снижаться, а градиент увеличиваться.

5. График зависимости  при 0.



На следующем графике видна погрешность:



*Справка*. В данных условиях тепловое нагружение отсутствует, причин для нагрева нет, температура стержня должна быть равна температуре окружающей среды  (разумеется, с некоторой погрешностью, определяемой приближенным характером вычислений).

# Вопросы при защите лабораторной работы

Ответы на вопросы дать письменно в Отчете о лабораторной работе.

1. **Какие способы тестирования программы можно предложить?**

Меняя коэффициенты, можно оценивать результат исходя из физического смысла задачи. Например, как было продемонстрировано в лабораторной, при F0 = 0 температура стержня должна быть равна температуре среды, а при отрицательных значениях идет съем тепла, температура должна увеличиваться.

**2. Получите простейший разностный аналог нелинейного краевого условия при**

где – заданная функция.

Производную аппроксимируйте односторонней разностью.

**(**

**3. Опишите алгоритм применения метода прогонки, если при краевое условие линейное (как в настоящей работе), а при**, **как в п.2.**

Будет использоваться правая прогонка.

(21)

Начальные прогоночные коэффициенты:

(22)

Прогоночные коэффициенты:

(23)

Зная, что , найдем :

В (20) подставим (21):

(24)

**4. Опишите алгоритм определения единственного значения сеточной функции в одной заданной точке . Использовать встречную прогонку, т.е. комбинацию правой и левой прогонок (лекция №8). Краевые условия линейные.**

Левая прогонка:

Прогоночные коэффициенты:

Объединив левую и правую (21), (23) прогонки

Получим систему:

Подставив второе уравнение в первое: